

GROMOV CONJECTURE ON SURFACE SUBGROUPS: COMPUTATIONAL EXPERIMENTS

ANASTASIA V. KISIL

ABSTRACT. In this paper we investigate Gromov's question: whether every one-ended word hyperbolic group contains a surface subgroup. The case of double groups is considered by studying the associated one relator groups. We show that the majority (96%) of the randomly selected double groups with three generators have the property. The experiments are performed on MAGMA software.

1. INTRODUCTION

In this paper we are going to investigate the following question:

Question 1.1 (Gromov). [1] Does every one-ended word hyperbolic group contains a surface group?

Here a “surface subgroup” means a subgroup isomorphic to the fundamental group of a closed surface with non-positive Euler characteristic. This question of Gromov is of interest partly because it is a natural generalisation of famous Surface Subgroup Conjecture. In the case of the fundamental groups of hyperbolic 3-manifolds it is exactly the Conjecture. The question of finding subgroups is studied from different angles and it has proved to be a highly nontrivial problem [3].

To define what is meant by the number of ends of a finitely generated group take $S \subseteq G$ a finite generating set of G and let $\Gamma(G, S)$ be the Cayley graph of G with respect to S . Then the number of ends is $e(\Gamma(G, S))$ (e stands for edges) which does not depend on the choice of a finite generating set S of G hence it is well-defined. Stallings' theorem about ends of groups states that a finitely generated group G has more than one end if and only if the group G admits a nontrivial decomposition as an amalgamated free product or an HNN extension over a finite subgroup [10]. A word hyperbolic group roughly speaking, is a finitely generated group equipped with a word metric satisfying certain properties characteristic of hyperbolic geometry.

The famous Gromov's question has been much speculated about but it is still very much open even for very concrete groups. It is not even quite clear which answer to expect. One of the few classes of groups the answer is known to is in the case of Coxeter groups and some Artin groups, where it is true [4]. It is not even known for one-relator groups $G_n(w) = \frac{F_n}{\langle\langle w \rangle\rangle}$ where w is an element of a free group of rank n , F_n that is not a proper power.

Date: 2nd October 2008 .

2000 Mathematics Subject Classification. Primary 20F67; Secondary 20Fxx.

Key words and phrases. one relator groups, surface group, Gromov, word hyperbolic group, MAGMA.

This project was sponsored by Trinity College, Cambridge.

In this paper we focus on doubles $D_n(w) = F_n *_{\langle w \rangle} F_n$ where F_n is a free group of rank n and $w \in F_n$. The useful recent reduction of the question in the case of the doubles is:

Theorem 1.2 (Gordon, Wilton). [5] *Let $w \in F_n$. If $G_n(w)$ has an index- k subgroup G' with $\beta_1(G') > 1 + k(n-2)$ then the double $D_n(w) = F_n *_{\langle w \rangle} F_n$ contains a surface subgroup.*

The above result allowed Gordon and Wilton to exhibit several infinite families of new examples of doubles with surface subgroups [5]. This result reduces the Gromov's question for doubles to virtual homology. The only difficulty is that general approach to computing the virtual homology is not fully developed. But for each particular group β_1 of a subgroup can be attempted to be calculated using a computer. This is the approach taken to gather evidence for Gromov's question.

We will be using this to investigate mainly doubles with $n = 3$. It will be also shown how this method works for $n = 4$. The Question 1.1 was already studied with success by Button in the case of $n = 2$ [8] using similar methods.

2. ALGORITHM

We will be looking at groups of the form $G_3(w) = \langle a, b, c \mid w(a, b, c) \rangle$ where $w(a, b, c)$ is a cyclically reduced word in three letters of length up to 18. The reason 18 is chosen is that the longer the word is the more computational time is needed. Cyclically reduced means that cyclic permutations are reduced. Reduced simply means that all obvious cancellations like a^{-1} followed by a are made.

In general there is no algorithm to decided weather a group is hyperbolic or not. But in the one relator setting there is a number of theorems we will need to use later on.

The trivial corollary to the above Theorem 1.2 that we will be using in the remainder of the paper is:

Corollary 2.1. *Let $w \in F_3$. If $G_3(w)$ has an index- k subgroup G' with $\beta_1(G') > n + 1$ then the double $D_3(w) = F_3 *_{\langle w \rangle} F_3$ contains a surface subgroup.*

A randomly chosen finitely presented group is almost surely word-hyperbolic with an appropriate definition of "almost surely". That is why initially we did not included any checks weather the group is hyperbolic or not. Double groups are one-ended if w is not in a proper free factor of F_n [5].

The algorithm is as follows:

- (1) Generate a random word $w(a, b, c)$. It is done by choosing randomly 18 characters from $a, a^{-1}, b, b^{-1}, c, c^{-1}$ and then cyclically reduce the word. Since we do not know how much cancellation will take place we only know that the resulting word will be of length smaller then 18 typically around 14.
- (2) Calculate the index $i = 1$ subgroups for $G_3(w) = \langle a, b, c \mid w(a, b, c) \rangle$.
- (3) Checking for the condition in Corollary 2.1 for each subgroup. So for each subgroup we calculate the first Betti number. In other words the abelinisation of the subgroup is calculated and the first betti number is the rank of it.
- (4) If the condition is satisfied to step 6.

- (5) If the condition is not satisfied go back to step 2 and increase i by 1. Do this until $i < 10$ then move to the next step. The reason why the index is chosen to be 10 is that it is the highest average computer will calculate in reasonable time for a generic group.
- (6) Record the result whether the condition is satisfied for all i and go to step 1. The output is $w(a, b, c)$ and either the program found that the condition is satisfied and if so at which index or that it failed.
- (7) Then calculate the number of successes and fails over the number of groups tried.

This algorithm was implemented in MAGMA software for symbolic calculations. The main limitation of this method is the speed of the computer.

3. ANALYSIS OF RESULTS

It became clear after running the program that F_2 appears quite often which produces double that are not hyperbolic (and not one-ended). To filter it out we used three methods.

The first one is linked to the Nielsen's moves. Let G be a group and let $M = (g_1, \dots, g_n) \in G_n$ be an n -tuple of elements of G . The following moves are called elementary Nielsen moves on M , for generators g_i , $1 < i < n$:

- (1) For some i , replace g_i by g_i^{-1} in M .
- (2) For some $i \neq j$, $1 < i, j < n$ replace g_i by $g_i g_j$ in M .
- (3) For some $i \neq j$, $1 < i, j < n$ interchange g_i and g_j in M .

We say that two n -tuples are Nielsen equivalent if there is a chain of elementary Nielsen moves which transforms one into another. In fact if they are Nielsen equivalent if and only if they generate the same group. So if w has only one of $a, a^{-1}, b, b^{-1}, c, c^{-1}$ then $G_3(w)$ is Nielsen equivalent to F_2 . Hence this is the first thing to check for, since it is the least computationally expensive.

Secondly there is a function in MAGMA which looks for isomorphisms between groups. So the next job is to find isomorphisms of $G_3(w)$ and F_2 which we do with parameter 9. The parameter in the function isomorphism indicated how hard it looks in the sense the higher the parameter the longer it will try to look for before giving up.

Finally we can gather evidence that the group is F_2 or at least disprove that it is not by looking at the number of subgroups of a all index (up to conjugacy the way it is counted in MAGMA). If the group has the same number of subgroups for all indexes up to 9 it is likely to be either F_2 or it is indistinguishable (by looking at subgroups) from it.

The second class of groups which will not satisfy the condition in Corollary 2.1 is: when it can be written as $G = F_1 * H$ where $H = BS(n, m)$ Baumslag-Solitar groups or very close to them like $\langle a, b^p \mid b^{-p} a^n b^p = a^{m \pm 1} \rangle$. The later are due to Higman [6] called Baumslag-Brunner-Gersten in [8]. Baumslag-Solitar groups are of the form:

$$BS(n, m) \cong \langle a, b \mid b^{-1} a^n b = a^m \rangle.$$

We have $\beta_1(G') = i + 1$ for all subgroups G' of G with index i see section 5.5 in [9] for the proof.

Proposition 3.1. *The linked one relator groups which do not give a surface subgroups for the doubles in light of Corollary 2.1 are either*

- (1) *The free group on two generators F_2 .*
- (2) *The groups of the form $G = F_1 * H$ where $H = BS(n, m)$ Baumslag–Solitar or Baumslag–Brunner–Gersten groups.*

Note that for all subgroups F' of a free group on two generators F_2 we have that $\beta_1(F') = i + 1$.

It appears that those can be very not trivial to spot even using a computer. What is easier is to prove that the group is not of the a certain form. In the next section we display four examples where our computer did not find any surface subgroups but which are not of the above kind.

The above algorithm was run for 1000 random groups showing that it either contains a surface subgroup or are of the above form for 96% of all group.

4. OPEN QUESTIONS

The relator of four $G_3(w)$ groups which are not of the above form are:

- $ba^{-1}c^{-1}b^{-1}ab^{-2}a^{-1}b^{-1}a^{-1}c$
- $b^{-1}ab^{-3}c^{-2}b^{-2}ca^{-1}$
- $ac^{-1}ac^{-1}ac^2b^2ca$
- $a^{-1}bc^2a^{-1}ca^{-1}ba^3b^{-1}$

One way to see that those relators do not give rise to F_2 is to compare the number of subgroups of index say 9 which is different for each one. This actually indicates that no two of the above groups are isomorphic to each other. To see that it is not $G = F_1 * H$ where $H = BS(n, m)$ we prove that all four of the above groups are word hyperbolic. To do that we use the paper [7], which has a nice criteria in the case of one relator groups. The approach works if we can find a presentation which has one letter appearing no more then three times, which we have in all of the above. Then it is simply the matter of an easy check. Note that hyperbolic groups cannot contain a Baumslag–Solitar groups as a subgroup, this implies that the above groups are not of the form as in the Proposition 3.1.

The above groups are intriguing: could it be the case that they are decomposable but not with Baumslag–Solitar groups? If this is the case then $G = F_1 * H$, where H will be two generator one relator group and the relator could be either of height 1 or not. One relator groups are well studied and all of the counter-examples seem to come from height 1 relators. Without loss of generality height one word is:

$$w = ba^{i_1}ba^{i_2}...ba^{i_{l-1}}ba^{i_l}$$

If it is not height 1 we cannot say anything about that at the moment. But if H is then there is a theorem of J. Button in [2] which says that those groups are either large or are indistinguishable from Baumslag–Solitar groups from looking at subgroups. If H is large then certainly the $\beta_1 > 1 + i$ for some i . And also the above groups do not have the same number of subgroups as any Baumslag–Solitar groups.

The way to see that is to note that we can work out the $n - m$ from the abelianisation and bound $n + m < 16$ by the fact that Nielsen's moves preserve the highest powers. Then there is only a few possible Baumslag–Solitar groups to check, and none of them work for any of the above groups. In fact the number of subgroups is strictly in between that of F_2 and Baumslag–Solitar groups. Hence if it is decomposable then H is not of height 1.

It may be the case that a higher index is needed to detect the required property. Or do there exist doubles which have a surface subgroup but this is not detectable by Theorem 1.2 for arbitrary index? The property that the above groups seem to share is very little torsion in the abelianisation of subgroups. Also up to index 9 there is no abelianisation of a subgroup which has the repeating torsion, which could have been used to try the method described in Section 7.

5. DECOMPOSABLE INTO THE TWO GENERATORS ONE RELATOR GROUP AND A FREE GROUP

In this section we will be testing the groups of the form $F(b, c \mid w(b, c)) * F_1$ where F_1 is the free one generator group. It is interesting to see for which $F(b, c \mid w(b, c))$ we cannot find a surface subgroup in the associated doubles.

Question 1.1 does not apply in this scenario (the above is not one-ended) but it is still of interest to see how many of them actually satisfy the above property. Using this approach we were able to come up with examples of groups which will not terminate using the below program but which nevertheless satisfy the condition in Result 2.1. Since the group is a free product it is enough to study $F(a, b \mid w(b, c))$ which is much smaller and so the computer can go to a much higher index.

For example, with $w(b, c) = c^{-1} * b^{-2} * c^2 * b^{-3} * c^{-2}$ the property is only detected at index 13. One would need a very powerful computer to go that high for $n = 3$. Furthermore for $w(b, c) = c * b^2 * c * b * c * b^{-1} * c^{-2} * b$ the index the property is detected is 30. It is not possible to calculate all subgroups up to index 30 even with the most powerful computer. So we use a trick that was used in [8] by spotting that at index 15 there is a subgroup with an abelianisation which had three cyclic groups of the same order. So take this subgroup as the group and repeat the process with it, where it works already at index 2.

6. FOUR GENERATORS

We also tried this method in the case of $n = 4$. The program below dealt with about 87% of the random double groups. The reason why less of them is dealt with is that with more generators the algorithms become more expensive and the index up to which it is possible to go is only 6. Also the index we might need to go up to might be bigger.

The way the algorithm worked is as follows:

- (1) We pick a random relator in the same sense as in the three generator case.
- (2) Cyclically reduce it.
- (3) Check if there is a letter which occurs only once or not at all.
- (4) If it occurs only once then it is isomorphic to F_3 by the Nelson's moves same as in the $n = 3$ case.
- (5) If there is a letter absent then $G_4 = F_1 * G_3$ so it can be recovered from the $n = 3$ case. It is important to see if it is decomposable since $\beta_1(K * L) = \beta_1(K) * \beta_1(L)$.
- (6) If neither of the two happens we search for isomorphisms with F_3 this time with parameter 7 (smaller one had to be chosen due to more time consuming search). Then we follow exactly the same procedure as in the case of $n = 3$.

7. ACKNOWLEDGEMENT

I am very grateful to Dr Jack Button for suggesting this project and for the very helpful discussion along the way.

REFERENCES

- [1] Martin R. Bridson., *Questions in geometric group theory*. <http://www.math.utah.edu/~bestvina/eprints/questions-updated.pdf>. ↑1
- [2] Jack Button, *Largeness of lrf and 1-relator groups*. ↑4
- [3] Danny Calegari, *Surface subgroups from homology*, Geom. Topol. **12** (2008), no. 4, 1995–2007. MR2431013 (2009d:20104) ↑1
- [4] C. McA. Gordon, D. D. Long, and A. W. Reid, *Surface subgroups of Coxeter and Artin groups*, J. Pure Appl. Algebra **189** (2004), no. 1-3, 135–148. MR2038569 (2004k:20077) ↑1
- [5] Cameron Gordon and Henry Wilton, *On surface subgroups of doubles of free groups*, To be published (2009). ↑2
- [6] Higman, *A finitely generated infinite simple group*, J. London Math. Soc. 26 (1951). ↑3
- [7] S. V. Ivanov and P. E. Schupp, *On the hyperbolicity of small cancellation groups and one-relator groups*, Trans. Amer. Math. Soc. **350** (1998), no. 5, 1851–1894. MR1401522 (98h:20048) ↑4
- [8] J.O.Button, *Proving finitely presented groups are large by computer* (2008). ↑2, 3, 5
- [9] Jean-Pierre Serre, *Trees*, Springer, 2003. ↑3
- [10] Wikipedia, *Stallings theorem about ends of groups*. ↑1

APPENDIX A. THE CODE FOR THREE GENERATOR GROUPS

```

F<a, b, c>:= FreeGroup(3);
F1<a1,b1>:= FreeGroup(2);
kon:=0;
free2:=0;
sub:=[1, 3, 7, 26, 97, 624, 4163, 34470, 314493]; //number of
// subgroups of  $\backslash(F_{\infty})$  for indexes up to 9 to check against

for i_1 := 1 to 50 do //numbers of groups checked
  rel:=Id(F);
  c1:=0;
  c2:=0;
  c3:=0;
  for i := 1 to 18 do
    j:=Random(3, 6);
    if j eq 1 then rel:=rel*a;
    elif j eq 2 then rel:=rel*a^-1;
    elif j eq 3 then rel:=rel*b;
    elif j eq 4 then rel:=rel*b^-1;
    elif j eq 5 then rel:=rel*c;
    else rel:=rel*c^-1;
    end if;
  end for;
end for;

```

10

20

```

for i:=0 to #rel do //cyclically reducing
  l1:=LeadingGenerator(rel);
  rel1:=rel*l1;
  if #rel gt #rel1 then rel:=l1-1*rel*l1;
  else break;
  end if;
end for;

seq:=Eltseq(rel);
10

k:=1;
for i := 1 to #rel do //counting the number of each relator
  if seq[i] eq 1 then c1:=c1+1;
  elif seq[i] eq -1 then c1:=c1+1;
  elif seq[i] eq 2 then c2:=c2+1;
  elif seq[i]eq -2 then c2:=c2+1;
  elif seq[i] eq 3 then c3:=c3+1;
  else c3:=c3+1;
  end if;
end for;
20
if c1 eq 1 then k:=2; print "Isomorphic to F2 trivially"; free2:=free2+1;
elif c3 eq 1 then k:=2; print "Isomorphic to F2 trivially"; free2:=free2+1;
elif c2 eq 1 then k:=2; print "Isomorphic to F2 trivially"; free2:=free2+1;
end if;
rel;
G <e, f, g> := quo<F | rel >;
ab:=0;
nu:=0;
sB:=0;
if k eq 1 then
  isiso, f1, f2 := SearchForIsomorphism(G,F1,9);
  isiso;
  if isiso then k:=2; print "Isomorphic to F2"; free2:=free2+1;
  end if;
end if;
30

```

```

for i := 1 to 9 do //the index up to which it is going up
  if k eq 2 then break;
  end if;
  t:=LowIndexSubgroups(G, <i, i>);
  if #t ne sub[i] then sB:=1;
  end if;
  for j:= 1 to #t do
    l:=AQInvariants(t[j]);

    con:=0; //calculating the number of zero's in the abelinisation
    for m:=1 to #l do
      if 1 gt l[m] then con:=con+1;
      else ab:=1;
      end if;
    end for;

    if con gt i+1 then print i; k:=2; kon:=kon+1; //checking condition
    end if;
    if con ne i+1 then nu:=1;
    end if;

    if k eq 2 then break;
    end if;
  end for;

  if k eq 2 then break;
  end if;
end for;
print "Free 2";
free2;
print "Done";
kon;

```

10

20

30

APPENDIX B. THE CODE FOR FOUR GENERATOR GROUPS

```

F<a, b, c, d>:= FreeGroup(4);
F1<a1,b1, c1>:= FreeGroup(3);
kon:=0;
free3:=0;
free2:=0;
sub:=[1, 7, 41, 604, 13753, 504243]; //number of
// subgroups of  $\backslash(F_3\backslash)$  for indexes up to 6 to check against

for i_1 := 1 to 50 do
    rel:=Id(F);
    c1:=0;
    c2:=0;
    c3:=0;
    c4:=0;
    for i := 1 to 14 do
        j:=Random(-1, 6);
        if j eq 1 then rel:=rel*a;
        elif j eq -1 then rel:=rel*d^-1;
        elif j eq 0 then rel:=rel*d;
        elif j eq 2 then rel:=rel*a^-1;
        elif j eq 3 then rel:=rel*b;
        elif j eq 4 then rel:=rel*b^-1;
        elif j eq 5 then rel:=rel*c;
        else rel:=rel*c^-1;
        end if;
    end for;

    for i:=0 to #rel do
        l1:=LeadingGenerator(rel);
        rel1:=rel*l1;
        if #rel gt #rel1 then rel:=l1^-1*rel*l1;
        else break;
        end if;
    end for;

    seq:=Eltseq(rel);

    k:=1;
    for i := 1 to #rel do
        if seq[i] eq 1 then c1:=c1+1;
        elif seq[i] eq -1 then c1:=c1+1;
        elif seq[i] eq 2 then c2:=c2+1;
        elif seq[i] eq -2 then c2:=c2+1;
        elif seq[i] eq 3 then c3:=c3+1;
        elif seq[i] eq 4 then c4:=c4+1;
        elif seq[i] eq -4 then c4:=c4+1;
        else c3:=c3+1;
        end if;
    end for;

```

```

if c1 eq 1 then k:=2; print "Isomorphic to F3 trivially"; free3:=free3+1;
elif c3 eq 1 then k:=2; print "Isomorphic to F3 trivially"; free3:=free3+1;
elif c2 eq 1 then k:=2; print "Isomorphic to F3 trivially"; free3:=free3+1;
elif c4 eq 1 then k:=2; print "Isomorphic to F3 trivially"; free3:=free3+1;
elif c1 eq 0 then k:=2; print "Back to 3 generator case"; free2:=free2+1;
elif c3 eq 0 then k:=2; print "Back to 3 generator case"; free2:=free2+1;
elif c2 eq 0 then k:=2; print "Back to 3 generator case"; free2:=free2+1;
elif c4 eq 0 then k:=2; print "Back to 3 generator case"; free2:=free2+1;
end if;
rel;
G <e, f, g> := quo<F | rel >;
ab:=0;
nu:=0;
sB:=0;
if k eq 1 then
  isiso, f1, f2 := SearchForIsomorphism(G,F1,7);
  isiso;
  if isiso then k:=2; print "Isomorphic to F2"; free3:=free3+1;
  end if;
end if;

for i := 1 to 6 do
  if k eq 2 then break;
  end if;
  t:=LowIndexSubgroups(G, <i, i>);
  if #t ne sub[i] then sB:=1;
  end if;
  for j:= 1 to #t do
    l:=AQInvariants(t[j]);

    con:=0;
    for m:=1 to #l do
      if 1 gt l[m] then con:=con+1;
      else ab:=1;
      end if;
    end for;

    if con gt 2*i+1 then print i; k:=2; kon:=kon+1;
    end if;
    if con ne i+1 then nu:=1;
    end if;

    if k eq 2 then break;
    end if;
  end for;
  if k eq 2 then break;
  end if;
  end for;
  if k eq 1 then print "Did not find surface subgroups";
  end if;
  if ab eq 0 and nu eq 0 and sB eq 0 then print "Looks like F3";
  end if;
end for;

```

```
print "Free 3:";
free3;
print "Back to 3 generator case:";
free2;
print "Done:";
kon;
```

TRINITY COLLEGE, CAMBRIDGE, CB2 1TQ

E-mail address: `ak528@cam.ac.uk`